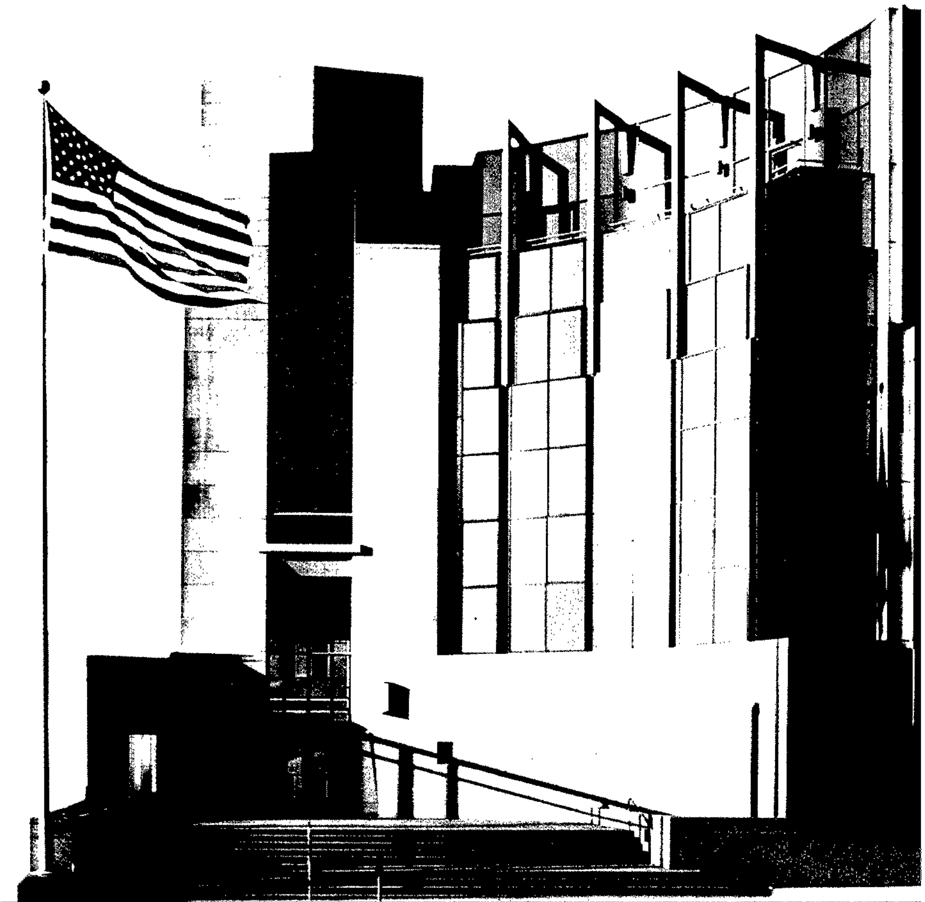
Carnegie Mellon
Software Engineering Institute

Software Product Lines: Experiences from the Eighth DoD Software Product Line Workshop

John K. Bergey
Sholom Cohen
Patrick Donohoe
Lawrence G. Jones

December 2005

TECHNICAL REPORT
CMU/SEI-2005-TR-023
ESC-TR-2005-023



Carnegie Mellon
Software Engineering Institute

Pittsburgh, PA 15213-3890

Software Product Lines: Experiences from the Eighth DoD Software Product Line Workshop

CMU/SEI-2005-TR-023
ESC-TR-2005-023

John K. Bergey
Sholom Cohen
Patrick Donohoe
Lawrence G. Jones

December 2005

Product Line Practice Initiative

This report was prepared for the

SEI Joint Program Office
ESC/XPK
5 Eglin Street
Hanscom AFB, MA 01731-2100

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

This work is sponsored by the U.S. Department of Defense. The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (http://www.sei.cmu.edu/publications/pubweb.html).

Table of Contents

Abstract

The Carnegie Mellon® Software Engineering Institute held the Eighth Department of Defense (DoD) Product Line Practice Workshop in September 2005. The workshop was a hands-on meeting to share DoD product line practices, experiences, and issues and to discuss ways in which specific product line practices are accomplished within the DoD. Participants reported encouraging progress on DoD software product lines. This report synthesizes the workshop presentations and discussions.

1 Introduction

1.1 Product Line Practice

A *software product line* is a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way [Clements 02a]. An increasing number of organizations are building their products as a product line in order to achieve large-scale productivity gains, improve time to market, maintain a market presence, compensate for an inability to hire, leverage existing resources, and achieve mass customization.

In January 1997, the Carnegie Mellon[®] Software Engineering Institute (SEI) launched the Product Line Practice Initiative to help facilitate and accelerate the transition to sound software engineering practices using a product line approach. The goal of this initiative is to provide organizations with an integrated business and technical approach to systematic reuse, so they can produce and maintain similar systems of predictable quality more efficiently and at a lower cost.

A key strategy for achieving this goal has been the creation of a conceptual framework for product line practice. The SEI *Framework for Software Product Line Practice*[SM] (henceforth referred to as "the framework") describes the foundational product line concepts and identifies the essential activities and practices that an organization must master before it can expect to successfully field a product line of software or software-intensive systems. The framework is a living document that is evolving as experience with product line practice grows. Version 4.0 is described in the book *Software Product Lines: Practices and Patterns*, and the latest version is available on the SEI Web site [Clements 02a, Clements 04].

The framework's contents are based on information-gathering workshops,[1] extensive work with collaboration partners, surveys and investigations, and continued research. The SEI has also incorporated practices reported at its international Software Product Line Conferences and collected from the community [Donohoe 00, Chastek 02, Nord 04, Obbink 05].

[®] Carnegie Mellon is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

[SM] Framework for Software Product Line Practice is a service mark of Carnegie Mellon University.

[1] The results of some of these workshops are documented in SEI reports [Bass 97, Bass 98, Bass 99, Bass 00, Clements 01].

In March 1998, the SEI hosted its first Department of Defense (DoD) product line practice workshop, *Product Lines: Bridging the Gap—Commercial Success to DoD Practice.* Topics discussed and documented included DoD barriers and mitigation strategies, and similarities and differences between DoD product line practice and commercial product line practices [Bergey 98]. Subsequent workshops were held in successive years [Bergey 99, Bergey 00a, Bergey 01, Bergey 03, Bergey 04a, Bergey 05]. At all seven DoD workshops, the SEI was encouraged to continue holding DoD workshops and to continue sharing best commercial and DoD practices through these forums.

One of the key outcomes of these workshops was the identification of product line practices that were particularly important to DoD acquisition organizations. This information supported development of a companion to the framework, titled *Software Product Line Acquisition: A Companion to A Framework for Software Product Line Practice* (henceforth referred to as "the companion") [Bergey 04b]. The companion, like the framework, is a living document with the latest version available on the SEI Web site at http://www.sei.cmu.edu /productlines/companion.html.

1.2 About This Workshop

The goals of the Eighth DoD Product Line Practice Workshop, held in September 2005, were to

- share DoD product line practices, experience, and issues regarding both development and acquisition
- discuss ways to motivate product line efforts in support of DoD systems
- explore ways to initiate software product line adoption in the DoD community

All participants in this workshop were from the DoD acquisition and contractor community. They were invited based on our knowledge of their experience with and commitment to software product lines as either DoD system acquirers or DoD system contractors. Together, we discussed the issues that form the backbone of this report.

The format of this workshop followed that of the previous successful seventh workshop. Participants were invited to make presentations about their organizations' activities and interests in software product lines. The small group size allowed extended discussion about each presentation. After the workshop, the group agreed once again that this format worked well.

The workshop participants included

- Ceci Albert, Acquisition Support Program, SEI
- John Bergey, Product Line Systems Program, SEI
- Grady Campbell, Acquisition Support Program, SEI
- Chi-Fang Chen, Boeing

- Bob Cohen, Program Manager, Force XXI Battle Command Brigade and Below (FBCB2)
- Sholom Cohen, Product Line Systems Program, SEI
- Patrick Donohoe, Product Line Systems Program, SEI
- Edward Dunn, U.S. Navy Naval Undersea Warfare Center (NUWC)
- Shuo Fang, Northrop Grumman
- Stephen Guine, Northrop Grumman
- Paul Jansen, Austin Info Systems
- Lawrence Jones, Product Line Systems Program, SEI
- Jim Linnehan, Office of the Assistant Secretary of the Army (Acquisition, Logistics, and Technology) (ASA [ALT])
- Reed Little, Product Line Systems Program, SEI
- Dan Vanderwarker, Aerospace Corporation
- Paul Work, Raytheon

1.3 About This Report

This document summarizes the presentations and discussions from the workshop. This report is written primarily for those in the DoD who are already familiar with product line concepts, especially those working on or initiating product line practices in their own organizations. Acquisition managers and technical software managers should also benefit from this report. Those who desire further background information are referred to the following publications:

- *Basic Concepts of Product Line Practice for the DoD* [Bergey 00b]
- *A Framework for Software Product Line Practice, Version 4.2* [Clements 04]
- *Software Product Line Acquisition: A Companion to A Framework for Software Product Line Practice, Version 3.0* [Bergey 04b]
- *Software Product Lines: Practices and Patterns* [Clements 02a]

The next section of this report contains a digest of the presentations. The report concludes with a brief summary.

2 DoD Software Product Line Experiences: A Digest of Participant Presentations

2.1 Introduction

Dr. Jim Linnehan,[2] of the Office of the Assistant Secretary of the Army (Acquisition, Logistics, and Technology), made some opening remarks to start the workshop. He believes that acquirers, suppliers, and end users of defense systems share a growing recognition that, especially given shrinking defense budgets, it makes no sense to build essentially the same systems over and over. Software product lines are increasingly seen as a way to address this problem.

He believes that the key challenges are not technical, but rather institutional, and that all three communities must work together to share ownership when adopting a product line approach. The key challenges include

- the need to develop life-cycle business justification for a product line approach to motivate organizations
- acquisition policies that encourage and enable a product line approach
- cultural changes and adjustments to the traditional rewards systems
- innovative and supportive contracting mechanisms
- organizational change to break down stovepipes
- education focused on architectures, capability engineering, systems of systems, and life-cycle management to enable software product lines

Dr. Linnehan believes that there are ways to address these challenging issues. He believes pilot programs, special projects, and forums like this workshop are good ways to develop realistic solutions and new approaches to make product line success easier to attain.

Following Dr. Linnehan's remarks, each workshop participant was given an opportunity to present and discuss his or her experiences with software product lines in the DoD environment. A summary of each presentation follows.

[2] Currently, Dr. Linnehan leads the U.S. Army Strategic Software Improvement Program (ASSIP) focused on improving the acquisition of software-intensive systems across the Army acquisition community.

2.2 Austin Info Systems and Software Product Lines - Paul Jensen

Paul Jensen of Austin Info Systems (AIS) presented an update of the work being done on the Joint Intelligence Toolkit (JIT), the technology base for a software product line of automated intelligence analysis products. The JIT provides a service-based architecture and components from which products are built for multiple customers in the DoD and the Department of Homeland Security. The toolkit has been developed from both legacy code and new development, with about a 50-50 split between these two categories. Components in the toolkit include "foundation" and existing components, AIS-developed products (e.g., AXIS and Medina), commercial off-the-shelf (COTS) products (e.g., RemoteView from Sensor Systems), and customer-specific components. There are about 120 software core assets totaling 1.3M source lines of code. The software runs on Windows, Linux, and Unix platforms.

The company faced some challenges in 2005, not least of which was balancing current customer deliverables with the move to the product line approach. The engineering department was reorganized to support the new approach; customers, however, remain leery of software product lines. The company was also acquired by Overwatch Systems and is now known as Overwatch Systems Tactical Operations (or TacOps for short) [Overwatch 05]. The plan is to integrate the JIT into the Overwatch Intelligence Center, a suite of integrated-intelligence collection and analysis capabilities. The customer base has become much broader since the company branched into the homeland security market.

Work in 2005 focused on changing the architecture to support software product lines and modifying current assets to fit into the changed architecture. The first delivery from the JIT occurred in August 2005. The ultimate goal is to have all programs use the JIT; current programs supported by AIS/TacOps include Counterintelligence Automated Tactical Exploitation and Information Software (CATEIS) for the Marine Corps, Future Combat Systems (FCS), and Distributed Common Ground Systems (DCGS).

Customer requests for products and features are examined by using a domain-modeling approach. AIS has created a domain model that is mapped to specific customer requests to determine coverage provided by assets. The model is not a completely "formal" one but has feature trees like most such models. The mapping to customer requests is currently a manual one. Some work has also been done on creating a production plan for targeted environments. So far no real metrics collection effort has been initiated to gauge the progress of the product line approach.

2.3 Reference Architectures, Software Product Lines and Model-Driven Development – Paul Work, Raytheon

Paul Work of Raytheon Integrated Defense Systems presented an overview of Leveraging Reference Architectures, Software Product Lines, and Model-Driven Development for Joint Battlespace Integration. This effort involves using a reference architecture as the starting point for the transition to software products lines and complementing it with a model-driven development (MDD) approach.

The goal is to implement a strategy to transition the organization from a clone-and-own reuse culture to a core asset software reuse culture. MDD was chosen as the approach to manage the inherent and accidental complexities of packaging subsets of application components into assemblies. (The number of components in large-scale distributed real-time and embedded systems, such as joint battlespace systems, ranges from hundreds to thousands.) The MDD paradigm includes the systematic application of domain-specific modeling languages as a basis for understanding and automating the development of complex systems. Raytheon is working with Vanderbilt University to develop a platform-independent component modeling language (PICML) to address the packaging problem in large-scale integration programs.

The overall approach for the joint battlespace integration effort is

1. Start with reference architectures.
2. Instantiate them with products from both Raytheon's suppliers' product lines and Raytheon's own product lines.
3. Use MDD approaches and tools to integrate the multiple components from these various product lines.

2.4 RangeWare – Ed Dunn, Naval Undersea Warfare Center

Ed Dunn provided an update on the status of the Naval Undersea Warfare Center (NUWC) product line to support Navy ranges. NUWC is a development lab component of the Naval Sea System Command (NAVSEA) – Division Newport. NUWC is the Navy's full-spectrum research, development, test and evaluation, engineering and fleet support center for submarines; autonomous underwater systems; and offensive and defensive weapons systems associated with undersea warfare [NUWC 05]. Dunn discussed progress made with product line asset enhancement and operations that had been discussed at previous DoD Workshops.

Dunn also discussed a new effort with implications for possible product line development. Currently, the NUWC product line organization is fielding a system that encompasses three operational subsystems at the Atlantic Undersea Test and Evaluation Center (AUTEC). Whereas previous systems in the product line addressed pieces within the full product line scope, the AUTEC system is integrating and deploying a full capability, approximately 40 applications in all. Roughly half of these are built from assets developed for earlier systems in the product line. The rest are new to the product line but will become assets for future systems. The AUTEC system is scheduled to begin parallel operations in January 2006. Parallel operations is a period of testing where both the new and old systems are run concurrently during normal operations.

At the Fifth DoD Product Line Workshop in 2002, Dunn presented a business case for new additions to the product line that projected savings of $6 million in development and $9.6 million in maintenance. Dunn reported that the hoped-for validation of the business case was undermined by delays or cancellation of key projects that were to have participated. (All factors leading to these decisions were unrelated to the product line approach.)

Dunn offered several lessons for validating business cases:

- Collect more than lines-of-code (LOC) metrics to estimate and track development efforts.
- Collect measures (e.g., LOC, defects, or level of effort) to the resolution of the subsystem or maybe even the component level. (Some systems report only for the entire software development effort.)
- Use a Goal-Question-Metrics (GQM) approach to determine the desired measures. Doing so will provide a basis for tracking measures based on business goals, which will support validation of the business case [Goethert 03].

Dunn addressed several challenges:

- institutionalizing the product line approach
- obtaining support, including recognition and funding, for the product line organization across all levels of command: local, NUWC, Navy sponsors and program offices, and future users of major facilities
- addressing the issue of what is desirable from an engineering point of view versus the constraints of available funding—what looks good in the future may not be possible now. (This leads to the principle of designing the product line so that capability may be added easily.)
- countering the following objections to product lines
  - "We already have a product line in place." [Short answer: What they often really have is multiple copies of the system, each morphed many times over. This is referred to as a "clone and own" approach, and unlike a product line approach, incurs maintenance costs for as many copies of the system as exist.]
  - "Our system already has the features we need, so we don't need the product line version." [Short answer: Maybe, but if the scope of the system expands (as is often the case) generating a new version of the system could be much cheaper as a member of a product line.]

- "The product line doesn't have a specific feature, so why follow product line?" [Short answer: Consider whether that feature might be incorporated into the product line or whether the benefits of that particular feature outweigh the other long-term savings of product line membership.

Despite the difficulties, the product line has scored some victories. One program saw, at first glance, little possibility of using the product line. However, when a legacy hardware system failed, a replacement was needed immediately. The program obtained the capability using new hardware, with control software built on the product line. The program received this replacement capability in a day and a half.

Dunn discussed systems for the Navy of the future following the "test, train, integrate" approach under the Navy after Next vision. An example of the dynamism the Navy seeks is the ability to set up a sea battle group en route. Deployed systems do not have the desired degree of agility to achieve this objective. Composability is the issue facing current systems that must be overcome to achieve this goal. The Navy needs a capability for dynamic composition. One example of this need occurs in the gathering of daily intelligence from multiple, diverse, and changing sources; given a unique question per day, the Navy must be able to explore available resources that can support the Commander's Critical Information Component. Dunn claims that such a system could be built on the concept of a product line of agents that can interact "on the fly" with services identified through the discovery mechanism of most service-based architectures. He and his team at NUWC, collaborating with other Navy offices, built and installed such a capability for several sea trials during 2004-05.

2.5 Advanced Multiplex Test System – Sholom Cohen, SEI

The Advanced Multiplex Test System (AMTS) program is producing a product line of system tools that support avionics test and analysis. The AMTS program falls under the Avionics Branch, Avionics/Intelligence and Electronic Warfare Division (A/IEW) of the Software Engineering Center (SEC) of the Army Communications-Electronics Life Cycle Management Command (CE LCMC). The tools support test and maintenance of U.S. DoD end items, including fully integrated, network-based avionics platforms and the individual constituent Line Replaceable Units (LRUs) that reside on the network. Platforms that implement networks based on MIL-STD-1553 include both aviation and ground vehicles maintained and operated by the U.S. DoD Joint Services and allied foreign military.

The vision for AMTS is to provide intuitive and interactive applications to assist diagnostic test and analysis personnel across all environments and maintenance levels (Unit, Intermediate, and Depot). Use of AMTS test and analysis products will support an efficient and accurate process to determine the health of an avionics network and its subcomponents. The AMTS program is building a common set of core assets as well as processes that prescribe how to produce test and analysis products from the assets. The AMTS production strategy is reactive,[3] building on legacy products that have been very successfully field tested to the satisfaction of DoD customers. New product-line-based products will update and replace the legacy systems.

A full AMTS product includes software, hardware, documentation, support materials, and training materials. The AMTS program is building assets in each of these areas for a broad scope of products. These products include test and analysis tools for

- Apache AH-64A & AH-64D helicopters
- Blackhawk HH-60L, UH-60M, and MH-60 helicopters
- Kiowa and Chinook helicopter variants
- limited fixed-wing aircraft and ground combat vehicles

All of these products use the MIL-STD-1553 bus architecture in network-specific military LRUs. AMTS products also support test and analysis of specific LRUs including a full range of avionics equipment.

The range of variation satisfied by the product line may be defined by a domain analysis that shows variations in a number of areas:

- bus type
    - 1553A
    - 1553B
    - Ethernet
    - 1773
    - ARINC (commercial aircraft)
- bus architecture
    - single bus
    - multiple bus
        - single type (e.g., 1553 only)
        - mixed type (e.g., 1553 & Ethernet)
    - nested (i.e., bus within bus; e.g., the Joint Tactical Radio System [JTRS])
- test level
    - Unit
    - Intermediate
    - Depot
    - Development (for test or verification and validation of systems under development)

[3] Krueger describes three product-line-adoption strategies: proactive (core assets are built before products), reactive (one or more products are built before the core asset base is established), or incremental (a combination of proactive and reactive) [Krueger 02].

Other variations include features for display types for messages; network topology display, and graphical aids; message handling: and modes (manual or automatic).

The major area of variation supported by the AMTS product line is for different field test levels: Unit, Intermediate, and Depot, as well as laboratory tests for items under development. These levels are defined as follows:

- Unit
  - platform or system tests intended to fault-isolate components (e.g., LRUs) and cable faults on aircraft
  - specialized tests/scenarios supporting user-defined needs
- Intermediate
  - LRU tests intended to provide go/no-go tests for system components outside assembled system (on-bench)
  - platform or system tests intended to assist modifications
  - platform or system tests "more comprehensive" than Unit
- Depot[4]
  - LRU and platform tests supporting maintenance work orders
  - LRU tests intended to fault-isolate to the PC card, subassembly, or (card) component level
  - platform or system tests supporting phase maintenance
- Development (for test or verification and validation of systems under development)

The AMTS product line will support this variation through a common configuration of software and hardware. This configuration includes

- client software to support user interfaces with database-driven customizations
- server software that interfaces with a database for message support and with the system under test
- hardware harnesses to link the AMTS computer either to the bus (for network testing at the Unit level) or directly to LRUs (for Intermediate/Depot testing)

The AMTS program is building the software assets and hardware links to provide support products across the entire scope of supported systems. The program's current schedule is to complete the first phase of asset development by the end of Calendar Year 2005. The AMTS program will use these assets to produce an initial product in the first quarter of Calendar Year 2006. During 2006, the program will upgrade the assets to improve production of current products and produce two or more additional products using the upgraded assets.

[4] Depots also have a need for Unit and Intermediate test capability.

The business case for successful AMTS development and operations is based on several assumptions:

- The optimum AMTS situation is a turn-key system (e.g., preloaded on maintenance-support-device [MSD] computers and ready to use), requiring no user set-up or configuration
- Field customers will receive a pre-configured, customized product containing only the features that interest them.
- Field customers should have to install new versions of AMTS products only when these releases are necessary to address new customer-specific requirements.
- Eventually, AMTS products should be delivered by contractors of Department of Army or other DoD test equipment/systems.[5] These contractors must practice an institutionalized product line approach.
- Customers are only interested in funding products of concern to them. They are also not interested in funding some cross-product capabilities for systems other than their own that share some common avionics.
- AMTS products also will be used in laboratories.

While the AMTS program does not have a quantified business case for its effort, it has a sufficient intuitive understanding of the payoffs to justify the approach. The program is currently collecting metrics to build a quantitative picture of the benefits.

2.6 Force XXI Battle Command Brigade and Below–Stephen Guine, Shuo Fang, Northrop Grumman

Stephen Guine and Shuo Fang reported on the progress of the Northrop Grumman Corporation of Dominguez Hills on Force XXI Battle Command Brigade and Below (FBCB2). FBCB2 is a tactical battle command system that fulfills two primary missions for maneuver troops:

1. situational awareness (SA), which includes the functions of
    a. blue force tracking (friendly-force position reports)
    b. enemy locations
    c. geo-referenced battlefield objects (e.g., minefields, obstacles, supply points)
    d. battlefield graphical overlays

[5] The program also has a goal of developing an organic product line capability within the government.

2. command and control (C2) via digital messaging between platforms, units, and higher echelon systems, which includes such functions as
   a. field orders
   b. spot reports
   c. status reports
   d. waypoints and navigation
   e. audio and visual message alerts
   f. auto-generated SA from C2 messages

Based on their initial experiences with building a demonstration data tablet product, Guine and Fang reported several lessons learned:

- Product line development requires a change in mind-set. You must guard against the tendency to focus on the development of products (the old mind-set) and switch to an emphasis on developing core assets. The staff must be taught to understand that the design and development of core assets is fundamentally different from that of single-product software. The concept of a core asset team as a supplier to product teams takes time to institutionalize.
- Agreement on the scope of the product line is critical. You must work carefully with the government customer to define the scope. Funding for an expanded, corporate scope must be identified and sought separately.
- Don't underestimate the need for formal training. Documentation and direct involvement of product line experts are not substitutes for effectively training the staff.
- Effective decision processes must be in place to allow technical decisions to be made quickly and then executed.

While there are several challenges still to be overcome, the presenters agreed that the product line approach offers substantial improvement and growth opportunities, making the up-front investment and commitment worthwhile.

2.7 Product Line Survey Results – Dan Vanderwarker, Aerospace

Daniel Vanderwarker of the Software Engineering Department of The Aerospace Corporation in Chantilly, Virginia, gave a presentation that addressed aspects of business and technical considerations for product line development. The discussion centered on a product line survey the corporation has periodically conducted and reported on at several of the annual Ground Systems Architecture Workshops (GSAWs). A Web-hosted version of the survey was disseminated to a large population of "buyers" and "developers." The survey results were useful for gathering insight into technical, business, and organizational considerations for product lines and investigated incentives and challenges concerning product line development (http://sunset.usc.edu/events/GSAW/gsaw2000/pdf/FinalVanderwarker.pdf). Surprisingly, 80% of the survey respondents answered from both the perspective of the buyer and developer (e.g., seller) while 15% of the respondents answered from the buyer perspective only. The survey involved assigning an importance rating to each of 32 software variables (survey items). Some of the more significant survey items were

- level of management commitment and support for strategic software reuse
- customer's preference to "build from scratch"
- degree of information sharing across programs/organizations
- level of developer experience with software reuse or product lines
- importance of buyer/developer interaction (this is underscored)

The survey emphasizes the importance of interaction and information sharing among organizations, to overcome cultural factors and differences in perspective. As a result, The Aerospace Corporation has participated in a business case initiative to gain additional insight into incentives and disincentives concerning strategic reuse and product line development from an organizational, cultural, and institutional perspective. Vanderwarker cited the product line work that was done at the National Reconnaissance Office (NRO) that included developing a business case for the Control Channel Toolkit (CCT) program,[6] which adopted a product line approach. Establishing a business case is an essential tool for combating the traditional "independent program management" culture that poses barriers and impediments to strategic software reuse and that results in stovepipes. Strong cultural pressures against reusing software and disincentives for this reuse include a reluctance to share knowledge or to relinquish control with respect to ownership issues.

Some important factors that are key to the success of a product line initiative include ensuring the permanence of championship and commitment; maintaining program continuity and momentum; providing a suitable organizational infrastructure (e.g., creating a "horizontal" technology integration office); and promoting an organizational culture that encourages innovation and rewards collaboration across organizational groups. Survey results show that horizontal integration and collaboration efforts can smooth the way for product line adoption within an organization.

2.8 Software Architecture Evaluation in DoD System Acquisitions – John Bergey, SEI

John Bergey's presentation was an outgrowth of his work in the SEI Product Line Systems Program to help DoD organizations adopt software architecture and product line practices.

[6] Clements describes the CCT project [Clements 02a].

His presentation covered

- the important role software architecture evaluation can play in reducing risk, whether in a traditional system acquisition or one using a product line approach
- how the SEI Architecture Tradeoff Analysis Method® (ATAM®) and SEI Quality Attribute Workshop (QAW) can be used effectively in a product line acquisition context
- how an ATAM evaluation and QAW can be proactively integrated, up front, into an RFP/contract, and the added importance of doing this in a product line acquisition
- how a DoD acquisition program recently applied these methods in acquiring a family of systems being developed with a product line approach

The Role of Architecture Evaluation

Since the software architecture of a software-intensive system greatly determines system quality, evaluating that architecture for fitness of purpose before the system is implemented (or undergoes a major modification) is a cost-effective approach to uncovering deficiencies and reducing risk. The software architecture for a product line is particularly important since it defines the variation points for the products in the product line. Clements provides an overview of software architecture and its importance [Clements 02b].

Architecture Evaluation Methods

The QAW and the ATAM are the basis for the advocated acquisition approach. Both the QAW and ATAM are highly collaborative methods that actively involve key system stakeholders [Kazman 03].

A QAW is usually conducted in the early stages of the system life cycle, before there is a software architecture, in order to

- ensure that affected stakeholders have a common understanding of the system's business/mission drivers
- elicit, collect, organize, and prioritize system-quality-attribute requirements

Once the software architecture has been developed, an ATAM-based evaluation can be conducted to bring stakeholders together to prioritize quality attributes, analyze architectural decisions, and identify architectural risks.

Integrating Architecture Evaluation in a System/Product Line Acquisition

Software architecture evaluations can be conducted *opportunistically* and performed under an existing contract at the request of a program manager. However, such opportunistic architecture evaluations may be problematic. Reactive evaluations often divert effort from the planned schedule of events and may require additional work of the contractors and system stakeholders. While uncovering risks may eventually save time and money, there is at least

® Architecture Tradeoff Analysis Method and ATAM are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

the appearance of impact on cost and schedule. Proactive software architecture evaluations, on the other hand, are *preplanned* and integrated in a request for proposal (RFP) for a system (or product line) acquisition. The benefit of a proactive approach is that all affected parties are on the same page from the outset of the RFP/contract, and the cost and schedule are known entities that are included as an integral part of each offeror's technical and cost proposals.

Although it is desirable to perform an architecture evaluation as early as possible in a system (or product line) acquisition, doing an evaluation as part of source selection can be highly problematic because it is not realistic to expect

- offerors to propose off-the-shelf software architecture solutions that will satisfy the government's requirements
- a government team to be able to evaluate multiple architectures under the stringent time and resource constraints that typify source selection
- a government team to be able to conduct multiple architecture evaluations without introducing any kind of variability that could lead to a potential protest

Conducting a proactive, early architecture evaluation after contract award is especially important in the acquisition of a product line. This is true because the architecture must be appropriately designed up front, with sufficient variation mechanisms to accommodate strategic reuse across a family of systems.

Incorporating architecture evaluations in an RFP/contract in the DoD requires appropriate language for

- Section C (Statement of Work [SOW])
- Section H (Special Contract Requirements)
- Section J (Contract Deliverables)
- Section L (Instructions to Offerors)
- Section M (Evaluation Factors for Award)

In Section J, the SOW simply reads, "An evaluation team shall conduct a series of software architecture evaluations in accordance with the special requirements of Section H." All the detailed requirements for conducting a QAW and an ATAM evaluation are embedded in Section H, in a form equivalent to what one would expect to find in a software architecture evaluation plan. This is necessary to ensure that the government's expectations are understood by all the offerors and that there is a common basis to plan and cost out the architecture evaluations. To do this the RFP/contract has to address such items as

- identifying participants in the architecture evaluations
- roles and responsibilities of the participants
- required training for evaluation team members
- staging of the software architecture evaluations
- transitioning of evaluation team responsibilities

- training course information and availability
- other rules of engagement (ROE)

Section L instructs offerors to summarize their software architecture evaluation approach in their technical proposals and show how the evaluations fit in with their risk management plan, project management plan, and integrated master schedule. Section J describes the associated Contract Deliverables that must include a software architecture description and a software architecture evaluation report. Section M, which elaborates on the specific evaluation criteria to be used to evaluate the technical reports, includes some factors relevant to adopting an architecture-centric approach and performing architecture evaluations.

Recent DoD System Acquisition Experience using the QAW and ATAM

One DoD acquisition program, Common Link Integration Processing (CLIP), successfully incorporated a QAW and a series of ATAM evaluations in its RFP/contract and is now fully engaged in the process of carrying them out following contract award. The CLIP program is a multi-service (Navy and Air Force), Acquisition Category (ACAT) II program that will

- provide a common message processing, gateway functionality, and platform interface
- integrate Tactical Data Links (TDLs) across platforms with a TDL requirement
- enable the transition of new and legacy platforms to a Network-Centric Warfare environment

The CLIP contract award was announced in June 2005 and the software architecture-related contractual events include

- completion of a QAW in July 2005
- delivery of a Software Architecture Description prior to the Critical Design review (CDR) in February 2006
- first ATAM evaluation in March 2006

Additional information on the CLIP program can be obtained at

http://www.washingtontechnology.com/news/1_1/outsourcing/26374-1.html and
http://www.irconnect.com/noc/press/pages/news_releases.mhtml?d=78914.

Summary

In summary

- Architecture evaluations should be included early in system acquisitions to reduce software acquisition risk.
- Commercial methods such as the QAW and ATAM can be used for these evaluative purposes and are consistent with acquisition reform principles.
- Proactively applying architecture evaluations during the contract performance phase works best.
- An approach (and corresponding contract language) for conducting software architecture evaluations was developed and applied in an actual DoD acquisition.

The approach that was presented has been applied successfully and can be tailored and adapted to meet the needs of other acquisition programs, regardless of whether these are acquiring a traditional system or a product line of systems.

3 Summary

The SEI's Eighth DoD Product Line Practice Workshop explored the product line practices of organizations in the DoD community by sharing experiences. This workshop demonstrated a continuance of the trend revealed during the most recent workshops: namely, software product line practice is becoming a reality in the DoD. All the presentations were based on experience rather than plans or speculation. Challenges and experience-based solutions were discussed.

As in previous workshops, the empirical and anecdotal evidence that the workshop participants brought to the discussion enhanced mutual understanding of the practices and issues as they apply to the DoD. Traditional DoD acquisition strategies are not naturally conducive to software product lines. However, it is clear that product line practice is possible within the DoD, and more DoD organizations are taking a product line approach.

In an effort to expand both the information base and the DoD community interested in software product lines, the SEI was encouraged by the participants to continue to hold similar workshops.

As before, the results of this workshop will be incorporated into the companion, which will continue to be refined and revised as the technology matures and as the SEI continues to receive feedback [Bergey 04b]. If you have any comments on this report or are using a product line approach in the development or acquisition of software-intensive systems for the DoD and would like to participate in a future workshop, please send email to Linda Northrop at lmn@sei.cmu.edu.

References

URLs are valid as of the publication date of this document.

[Bass 97] Bass, L.; Clements, P.; Cohen, S.; Northrop, L.; & Withey, J. *Product Line Practice Workshop Report* (CMU/SEI-97-TR-003, ADA327610). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1997. http://www.sei.cmu.edu/publications/documents/97.reports/97tr003/97tr003abstract.html.

[Bass 98] Bass, L.; Chastek, G.; Clements, P.; Northrop, L.; Smith, D.; & Withey, J. *Second Product Line Practice Workshop Report* (CMU/SEI-98-TR-015, ADA354691). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1998. http://www.sei.cmu.edu/publications/documents/98.reports/98tr015/98tr015abstract.html.

[Bass 99] Bass, L.; Campbell, G.; Clements, P.; Northrop, L.; & Smith, D. *Third Product Line Practice Workshop Report* (CMU/SEI-99-TR-003, ADA361391). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1999. http://www.sei.cmu.edu/publications/documents/99.reports/99tr003/99tr003abstract.html.

[Bass 00] Bass, L.; Clements, P.; Donohoe, P.; McGregor, J.; & Northrop, L. *Fourth Product Line Practice Workshop Report* (CMU/SEI-2000-TR-002, ADA375843). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000. http://www.sei.cmu.edu/publications/documents/00.reports/00tr002.html.

[Bass 03] Bass, L.; Clements, P.; & Kazman, R. *Software Architecture in Practice,* 2nd ed. Reading, MA: Addison-Wesley, 2003.

[Bergey 98] Bergey, J.; Clements, P.; Cohen, S.; Donohoe, P.; Jones, L.; Krut, R.; Northrop, L.; Tilley, S.; Smith, D.; & Withey, J. *DoD Product Line Practice Workshop Report* (CMU/SEI-98-TR-007, ADA346252). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1998. http://www.sei.cmu.edu/publications/documents/98.reports/98tr007/98tr007abstract.html.

[Bergey 99] Bergey, J.; Campbell, G; Clements, P.; Cohen, S.; Jones, L.; Krut, R.; Northrop, L.; & Smith, D. *Second DoD Product Line Practice Workshop Report* (CMU/SEI-99-TR-015, ADA375845). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1999. http://www.sei.cmu.edu/publications/documents/99.reports/99tr015/99tr015abstract.html.

[Bergey 00a] Bergey, J.; Campbell, G; Cohen, S.; Fisher, M.; Jones, L.; Krut, R.; Northrop, L.; O'Brien, W.; Smith, D.; & Soule, A. *Third DoD Product Line Practice Workshop Report* (CMU/SEI-2000-TR-024, ADA387259). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000. http://www.sei.cmu.edu/publications/documents/00.reports/00tr024.html.

[Bergey 00b] Bergey, J.; Fisher, M.; Gallagher, B.; Jones, L.; & Northrop, L. *Basic Concepts of Product Line Practice for the DoD* (CMU/SEI-2000-TN-001, ADA375859). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000. http://www.sei.cmu.edu/publications/documents/00.reports/00tn001.html.

[Bergey 01] Bergey, J. et al. *Fourth DoD Product Line Practice Workshop Report* (CMU/SEI-2001-TR-017, ADA399205). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2001. http://www.sei.cmu.edu/publications/documents/01.reports/01tr017.html.

[Bergey 03] Bergey, J. et al. *Fifth DoD Product Line Practice Workshop Report* (CMU/SEI-2003-TR-007, ADA416733). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2003. http://www.sei.cmu.edu/publications/documents/03.reports/03tr007.html.

[Bergey 04a] Bergey, J.; Cohen, S.; Jones, L.; & Smith, D. *Software Product Lines: Experiences from the Sixth DoD Software Product Line Workshop* (CMU/SEI-2004-TN-011). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2004. http://www.sei.cmu.edu/publications/documents /04.reports/04tn011.html.

[Bergey 04b] Bergey, J.; Campbell, G; Cohen, S.; Fisher, M.; Gallagher, B.; Jones, L.; Northrop, L.; & Soule, A. *Software Product Line Acquisition: A Companion to A Framework for Software Product Line Practice, Version 3.0.* http://www.sei.cmu.edu/productlines /companion.html (2004).

[Bergey 05] Bergey, J.; Cohen, S.; Donohoe, P.; & Jones, L. *Software Product Lines: Experiences from the Seventh DoD Software Product Line Workshop* (CMU/SEI-2005-TR-001). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2005. http://www.sei.cmu.edu/publications/documents /05.reports/05tr001.html.

[Brownsword 96] Brownsword, L. & Clements, P. *A Case Study in Successful Product Line Development* (CMU/SEI-96-TR-016, ADA315802). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1996. http://www.sei.cmu.edu/publications/documents /96.reports/96.tr.016.html.

[Chastek 02] Chastek, G, ed. *Proceedings of the Second Software Product Lines Conference (SPLC2)* (Lecture Notes in Computer Science [LNCS] volume 2379). San Diego, CA, Aug. 19-22, 2002. New York, NY: Springer-Verlag, 2002.

[Clements 01] Clements, P. et al. *Fifth Product Line Practice Workshop Report* (CMU/SEI-2001-TR-027, ADA272355). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2001. http://www.sei.cmu.edu/publications/documents /01.reports/01tr027.html.

[Clements 02a] Clements, P. & Northrop, L. *Software Product Lines: Practices and Patterns*. Reading, MA: Addison-Wesley, 2002.

[Clements 02b] Clements, P. *Software Architecture: Key to Software System Quality*. http://www.sei.cmu.edu/ata/key/key_slide1.htm (2002).

[Clements 04] Clements, P. & Northrop, L. *A Framework for Software Product Line Practice, Version 4.2.* http://www.sei.cmu.edu/productlines /framework.html (2004).

[Cohen 02]	Cohen, S.; Dunn, E.; & Soule, A. *Successful Product Line Development and Sustainment: A DoD Case Study* (CMU/SEI-2002-TN-018, ADA407788). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2002. http://www.sei.cmu.edu /publications/documents/02.reports/02tn018.html.
[Cohen 04]	Cohen, S.; Zubrow, D.; & Dunn, E. *Case Study: A Measurement Program for Product Lines* (CMU/SEI-2004-TN-023). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2004. http://www.sei.cmu.edu/publications/documents /04.reports/04tn023.html.
[Donohoe 00]	Donohoe, P., ed. *Software Product Lines: Experience and Research Directions, Proceedings of the First Software Product Line Conference (SPLC1).* Denver, CO, Aug. 28-31, 2000. Boston, MA: Kluwer Academic Publishers, 2000.
[Goethert 03]	Goethert, W.; & Fisher, M. *Deriving Enterprise-Based Measures Using the Balanced Scorecard and Goal-Driven Measurement Techniques* (CMU/SEI 2003-TN-024, ADA418401). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2003. http://www.sei.cmu.edu/publications/documents /03.reports/03tn024.html.
[Kazman 03]	Kazman, R.; Nord, R. & Klein, M. *Life-Cycle View of Architecture Analysis and Design Methods (*CMU/SEI-2003-TN-026, ADA421679) Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2003. http://www.sei.cmu.edu /publications/documents/03.reports/03tn026.html.
[Krueger 02]	Krueger, C. W. "Easing the Transition to Software Mass Customization," 282-293. *Proceedings of the 4th International Workshop on Software Product Family Engineering*. Bilbao, Spain, Oct. 3-5, 2001. Berlin, Germany: Springer-Verlag, 2002.
[Nord 04]	Nord, R., ed. *Proceedings of the Third International Software Product Lines Conference (SPLC 2004)* (Lecture Notes in Computer Science [LNCS] volume 3154). Boston, MA, Aug. 3-Sept. 2, 2004. New York, NY: Springer-Verlag, 2004.
[NUWC 05]	*Naval Undersea Warfare Center.* http://www.nuwc.navy.mil/ (2005).

[Obbink 05] Obbink, H.; & Pohl, K. *Software Product Lines, Ninth International Conference, SPLC 2005 Proceedings*, (Lecture Notes in Computer Science [LNCS] volume 3714). Rennes, France, Sept. 26-29, 2005. Berlin Heidelberg, Germany: Springer-Verlag, 2005.

[Overwatch 05] Overwatch Systems, Tactical Operations: http://tactical.overwatch.com (2005).

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE December 2005	3. REPORT TYPE AND DATES COVERED Final

4. TITLE AND SUBTITLE

Software Product Lines: Experiences from the Eighth DoD Software Product Line Workshop

5. FUNDING NUMBERS

FA8721-05-C-0003

6. AUTHOR(S)

John K. Bergey, Sholom Cohen, Patrick Donohoe, Lawrence G. Jones

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

8. PERFORMING ORGANIZATION REPORT NUMBER

CMU/SEI-2005-TR-023

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

HQ ESC/XPK
5 Eglin Street
Hanscom AFB, MA 01731-2116

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

ESC-TR-2005-023

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified/Unlimited, DTIC, NTIS

12B DISTRIBUTION CODE

35

13. ABSTRACT (MAXIMUM 200 WORDS)

The Carnegie Mellon® Software Engineering Institute held the Eighth Department of Defense (DoD) Product Line Practice Workshop in September 2005. The workshop was a hands-on meeting to share DoD product line practices, experiences, and issues and to discuss ways in which specific product line practices are accomplished within the DoD. Participants reported encouraging progress on DoD software product lines. This report synthesizes the workshop presentations and discussions.

14. SUBJECT TERMS

DoD product line practice, Framework for Software Product Line Practice, software product line, software product line acquisition

15. NUMBER OF PAGES

35

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL